Computer Networks 157 (2019) 68–77

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Heterogeneous slot scheduling for real-time industrial wireless sensor networks

Changqing Xia [a,b,c], Xi Jin [a,b,c], Linghe Kong [a,d], Chi Xu [a,b,c], Peng Zeng [a,b,c,*]

[a] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China
[b] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences Shenyang, China
[c] Key Laboratory of Networked Control System, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China
[d] Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Article history:
Received 15 November 2018
Revised 11 April 2019
Accepted 12 April 2019
Available online 18 April 2019

Keywords:
Industrial networks
Scheduling
CPS
Real-time
Heterogeneous

ABSTRACT

Real-time performance and reliability are two critical indicators in an industrial wireless sensor network (IWSN). Several time-division multiple-address (TDMA)-based industrial standards such as WirelessHART and ISA100 are widely used in IWSNs. However, to simplify the analysis, standard TDMA supports only one or two slot types in each frame, and each slot is usually 10 ms, which severely limits transmissibility and real-time responses in TDMA-based IWSNs where the number of transmissions is large but the length of most packets is small. In this paper, we propose a TDMA frame containing slots of different lengths to address this issue. The key ideas are to waste fewer slot resources and achieve on-demand slot allocation. First, we study the matching problem in a TDMA frame; then, we propose two scheduling methods, the split scheduling algorithm (SSA) and the double plug-in algorithm (DPA), under our TDMA frame. Extensive simulations and real testbed results show that the proposed solution DPA can significantly improve network performance and reliability. Real-time comparisons with other existing scheduling schemes show that the proposed solution improves the acceptance ratio by 48.8% compared to the rate-monotonic scheme.

1. Introduction

As the technical foundation of intelligent manufacturing, an industrial wireless sensor network (IWSN) offers solutions that can transform the operations and roles of many existing industrial systems. Using IWSNs, industrial systems can achieve flexible control and customized production [1]. The sensing information and control messages are transmitted wirelessly between machines [2,3]. However, errors or disasters may occur when data are transmitted via an unreliable connection or when the data arrives late, which is a primary challenge for IWSNs. Several industrial standards have been proposed to address the high reliability and real-time requirements in IWSNs, such as WirelessHART, ISA100 and WIA-PA [4–6]. Although the existing industrial standards and scheduling methods have been considered in reliable, real-time wireless transmission situations [7,8], the existing methods do not meet the strict requirements of some IWSNs, such as when system demand exceeds the available resources [9].

Considering that the lengths of most transmitted data are small in IWSNs, we adopt a frame that can allocate slots of different lengths to different flows as needed to solve the aforementioned problems (some previous works have also studied variable-length slots in the TDMA protocol, however, they focused only on low-energy or bandwidth overhead and did not address real-time performance and reliability). In this approach, many slots with different lengths exist in one frame. The length of a slot is determined by the size of the packet. In each frame, the number of slots is increased. However, the system still cannot be scheduled accurately because of delays caused by transmission conflicts and channel contention. Consequently, we study scheduling approaches for heterogeneous-slot IWSNs. There are two main challenges in the heterogeneous slot approach: (1) matching node transmissions with different slot sizes and (2) allocating network resources to improve schedulability. If the wait time of a packet with small slots is the same as that of a regular slot, it will have the same performance as the system with regular slots. To address these two challenges, we propose a series of methods and provide several results. The contributions of this paper are as follows:

1. In many situations, a packet does not need to use the entire capacity of frame to transmit instructions; thus, we de-

* Corresponding author at: Shenyang Institute of Automation, Chinese Academy of Sciences, Nanta Street, NO. 114. Shenhe District, Shenyang 110016, China.
E-mail addresses: xiachangqing@sia.cn (C. Xia), zp@sia.cn (P. Zeng).

https://doi.org/10.1016/j.comnet.2019.04.005
1389-1286/

sign a frame model containing heterogeneous slots to increase the network resources (slot resources) for real-time IWSNs. The length of a slot is determined by the size of the packet - an approach that allows us to increase the slot resources in an IWSN. Compared with existing works on variable-slot or optimal-slot schemes, our model achieves on-demand allocation and allows multiple slots with different length to coexist in an IWSN.

2. We propose the algorithm of transmission matching (ATM) to pair the sending and receiving nodes for each transmission under our heterogeneous-slot IWSN. Furthermore, we prove that system performance is independent from traditional slot partitioning.
3. We propose SSA (split scheduling algorithm) and DPA (double plug-in algorithm) to improve the performance of IWSNs based on our heterogeneous-slot model, and we obtain the necessary and sufficient conditions for performance improvement.

The remainder of the paper is organized as follows. Section 2 reviews the research on real-time IWSNs. System configuration and problem statement is presented in Section 3. We study transmission matching algorithm in Section 4. The scheduling algorithms for heterogeneous slots are proposed in Sections 5 and 6. Section 7 illustrates the implementations of DPA based on both simulations and practical test. Conclusions is given in Section 8.

2. Related work

In recent years, IWSN applications have become increasingly widespread as researchers have focused on how to use and improve the reliability and real-time performance of IWSNs [10,11]. The author of Willig [12] first provided a deep discussion concerning the use of wireless technology in industrial automation; readers are encouraged to consult this reference for a deeper understanding of the field, several wireless industrial applications have been proposed, e.g., [13–16].

Because delays caused by transmission conflicts and channel contention greatly affect performance, some researchers have begun to concentrate on transmission scheduling in IWSNs [17–19]. The scheduling problem in WSNs is to determine the smallest length conflict-free assignment of slots, which has been proved as a NP-complete problem by Ergen and Varaiya [20]. For IWSNs, Saifullah et al. formulated a real-time transmission scheduling problem based on the characteristics of WirelessHART networks and proved that it is NP-hard [21]. Moreover, they analyzed the end-to-end communication delay in industrial wireless networks in [22]. Based on these works, Saifullah et al. analyzed transmission schedulability under a graph routing approach in WirelessHART networks. This method can be used to quickly assess the schedulability of real-time flows that have stringent reliability and latency requirements. In addition, in [23], a hierarchical data transmission framework was proposed that integrates the advantages of these schemes and makes a tradeoff among real-time response, reliability, and scalability. Wenchen Wang et al. proposed a dynamic network scheduling solution to minimize errors in system control applications by considering the application behavior and changing its priority based on dynamic conditions. In [24], the author proposed a novel method based on the segmented slot assignment, fast slot competition, and free node concept to improve the retransmission efficiency for time-division-multiple-access-based multihop IWSNs by using limited shared slot resources more efficiently. The challenges and opportunities in IWSNs are summed up by Sisinni in [25].

Some works have focused on the effective relationship between network resources and system performance. By analyzing the worst-case delay caused by transmission conflict and channel contention, Xia et al. [26] obtained the system demand bound function of mixed-criticality industrial systems, which can quickly determine system schedulability. Lu et al. [27] empirically analyzed the impact of channel selection on network topology, routing, and scheduling on a 52-node wireless sensor and actuator network (WSAN) testbed.

Some works have focused on finding an optimal TDMA slot size. The authors of Zhang and Gburzynski [28] proposed a variable-length slot time-division multiple-access (TDMA) protocol for personal communication systems aimed at accommodating sessions with diverse patterns and priorities between a node and the mobile station. Hussain et al. proposed low energy adaptive slot allocation that replaced the fixed slot size in classical TDMA schemes with a variable slot size that dynamically adapted to the data size generated at sensor nodes [29]. A more current approach [30] allows multiple WIFI transceivers to transmit simultaneously by synchronizing the access points across the network and using scheduled TDMA; in this manner, the central controller can schedule more or longer transmission slots for one AP than for another AP during any time interval. In [31], the author studied optimal TDMA time slot and cycle length allocation for hard real-time systems and presented an analytic method to determine the provably smallest possible slot length that must be allocated to a TDMA resource with a fixed cycle length and bandwidth. This method served a hard real-time load with arbitrary deterministic timing behavior, however, the proposed method only optimized the parameter configuration and lacked optimization of the scheduling method. Overall, the current works cannot fully exploit the power of IWSNs. Hence, there is an urgent need to investigate how to improve the real-time performance and reliability of IWSNs using on-demand heterogeneous slots.

3. System configuration

3.1. Network model

Many different data types coexist in IWSNs, such as alarms and sensor data. To guarantee that IWSNs remain reliable and respond in real time, these data should all be transmitted to their destination within specific timeframes. However, that performance level cannot be guaranteed in some situations. For example, a task may miss its deadline when the transmission time is longer than its period. Hence, we design a heterogeneous-slot structure that contains slots of different lengths (which will be explained later). The model consists of a gateway, a centralized controller and some wireless nodes. Furthermore, our design is based on state-of-the-art industrial network standards such as WIA, ISA100 and WirelessHART and has the following salient features: (1) a limited network size; (2) an IEEE 802.15.4 physical layer that allows per-time-slot channel hopping and (3) a MAC layer running a multichannel TDMA protocol [4]. The nodes in our design are equipped with a half-duplex omnidirectional radio transceiver that alternates the node status between transmitting and receiving; thus, our design has the same problems of transmission delays caused by transmission conflicts and channel contention as does a regular wireless industrial system [32]. Furthermore, our system can adopt the existing methods such as [33–35] to avoid the noise caused by multi channls.

In our design, wireless nodes constitute a multihop wireless mesh network. All sensory data are forwarded to a single centralized controller destination. The set of wireless nodes is denoted as $N = \{n_1, n_2 \ldots n_e\}$, where $e = |N|$ is the number of nodes in N, and the centralized controller is denoted as n_0. We define an end-to-end data transmission between a source and the centralized controller as a *flow*; the set of flows in the network is denoted by $F = \{f_1, f_2 \ldots f_z\}$, where $z = |F|$ is the number of flows in F. The

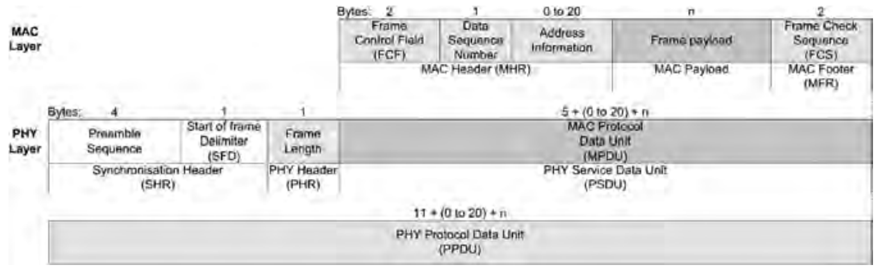


Fig. 1. Schematic view of the IEEE 802.15.4 frame format.

characteristics of a flow j in our system are denoted by $\{c_j, t_j, d_j\}$, where the number of transmission hops is c_j, and each slot can transmit one hop, that is, the flow requires at least c_j slots to reach its destination. The period is t_j, and the transmission limit is d_j. Hence, we can describe any flow j as follows. The source generates packets for the flow with a period of t_j, and these packets travel for c_j hops on the routing path (we do not discuss flow paths because our algorithms do not change them). Each packet is forwarded to its destination before the time limit d_j. In addition, all packets generated by the flows transmit under a multichannel TDMA protocol. There are 16 nonoverlapping channels in our physical layer, which is identical to the IEEE 802.15.4 protocol (e.g., $m = 16$). Furthermore, all flows in this study are periodic tasks (where the arrival time interval is equal to the period; that is, for any flow, the packet arrives at the source node at the beginning of each period).

For regular IWSNs scheduled under the TDMA protocol, each transmission slot is 10 ms. Because the IEEE 802.15.4-compliant radios commonly used in wireless sensor networks (WSNs) have a maximum bandwidth of only 250 kbit/s, each slot can transmit approximately 312 bytes ($\frac{250\times1000}{100\times8} = 312\,\text{Bytes}$). Fig. 1 depicts the IEEE 802.15.4 frame format [36], and at most 307 bytes of memory space are available to store sensor data (at least 287 bytes of memory space when address information is taken into account). For some types of data, such as alarms and control information, 307 Bytes is too large and will waste resources. Hence, we propose a heterogeneous-slot structure to improve the performance of IWSNs. We denote packets that cannot be scheduled and that do not need to be transmitted within their regular slots as small packets and denote the corresponding slot in the transmission-small packet as a small slot. Based on the size of the small packet, we split the regular slot (for example, the regular slot is 10*ms* in WirelessHART) into a small slot to achieve on-demand performance levels for each packet. The small slot is denoted as l, where $l = (\frac{dps\times10}{307} + hd)$ms, dps is the size of the small packet, and hd is the TDMA header, which is also contained ACK, CCA and the other fuctions. When the traditional 10 ms slot is denoted as L, we can obtain an on-demand parameter that reflects the proportion of real-time and traditional slots and is expressed as $\beta = \frac{l}{L}$. The flow sets that transmit in the regular and small slots are denoted as $F^L = \{f_1^L, f_2^L, \dot{f}_i^L\}$ and $F^l = \{f_1^l, f_2^l, \dots f_j^l\}$, respectively. In addition, in our system, the frame is defined as the lowest common multiple of all flow periods, which are denoted as T. The unit of a T is slots. There are two flows, the periods are 3 and 4. The frame is $T = 3 \times 4 = 12$.

Our heterogeneous-slot structure prioritizes allocations using a rate-monotonic (RM) scheduling scheme, which is the state-of-the-art technique for the allocation of flow priority in real-time systems and industrial wireless sensor networks [37] (we can also use the other scheduling policies, such as fixed priority and earliest deadline first instead of RM). In this scheme, a flow's priority is allocated based on its period. For example, for flows p and q, the priority of flow q is lower than that of flow p when $t_q > t_p$. Obviously, adjusting the slot length does not affect priority. Furthermore, the flow with smaller deadline has higher priory when two flows have the same period in RM.

Because both reliability and real-time response are important criteria in industrial networks, we use system schedulability to evaluate both reliability and real-time response. When the system is schedulable, we say the system is reliable and can meet its real-time requirement. In addition, the reliability depends on how many packets are received non-corrupted at the receiver side in some applications. However, each packet in industrial system has its own meaning. In this work, we say the system is reliable when all the packets can reach their destinations. The definition of schedulable is as follows:

Definition 1. schedulable: The flow set is schedulable if all flows can meet their deadlines.

It worth mentioning that our model may reduce the percentage of workload in each regular slot because some other functions (such as ACK, CCA, synchronization time and so on) are also executed in one slot. In this work, we assume all these additional functions are parts of the workload in the packets transmitted under small slots.

3.2. Problem statement

Given an industrial network, the flow set F, and the RM scheduling policy, our objective is to improve the schedulability of the IWSN by adjusting the sizes of slots for the flows that cannot be scheduled. To analyze the feasibility and characteristics of our heterogeneous-slot IWSN, we study only situations in which there are two types of slots in each frame. That is, the proportion of real-time and traditional slots is $\beta = \frac{l}{L} = 0.5$. Our model can easily be extended to systems with more slot types that satisfy $\beta^w = \beta_1\beta_2\dots\beta_i\dots\beta_w$, where $\beta_i = \frac{l_i}{L}$; l_i is the small slot allocated to f_i. There are several challenges in this work:

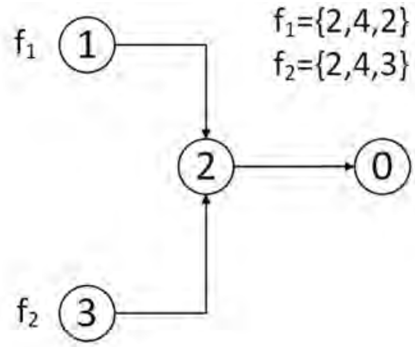
- Because the IWSN provides reliability and real-time transmission based on a multichannel TDMA protocol, the approach for matching node transmission with different slot sizes is a great challenge in our system.
- When each node is matched with the nodes one hop away regarding its transmission slots and channels, the approach for allocating network resources to improve schedulability is another challenge. If the packet that transmits under a small slot needs to wait the same amount of time as a packet with a regular slot, it will have the same performance as a system with only a regular slot.

The transmission matching is defined as follows:

Definition 2. Transmission matching: We define transmission matching for the IWSN as any node i in the network that knows when it should receive or send a packet through which channel and from or to which node.

	receive slot	send slot
node 1		1
node 2	1	3
node 3	4	

Fig. 2. An example for unmatched transmission.



(a) Routing,

f_1	1-2	2-0		
f_2			3-2	miss

(b) Transmission slots under the regular slot,

f_1	1-2	1-2	2-0	2-0				
f_2					3-2	2-0		

(c) Transmission slots under the small slot,

Fig. 3. An example of a TM.

4. Transmission matching

In an IWSN, sensors communicate through low-power multihop wireless mesh networks. By allocating slots and channels to each link, IWSN can provide a guaranteed delay transmission (the end-to-end delay is predictable). Because an oversized communication delay may degrade the system's performance or cause system errors, we optimize network resource allocations (transmission slots and channels) for the flows before the IWSN operates. However, the IWSN cannot work when the nodes are unable to find a match with the nodes one hop away on its transmission slots and channels; in other words, transmission matching is the foundation of reliable communication in IWSN. Hence, we first study transmission matching before addressing centralized scheduling.

Fig. 2 shows an example when a transmission cannot be matched. We assume that only one channel exists in this example; thus, nodes 1 and 2 can communicate because node 2 works in the receiving mode when node 1 sends the packet. However, node 3 cannot receive a packet from node 2 because their transmission slots are not matched.

As noted previously, the analysis of transmission matching starts at $\beta = 0.5$. When the system cannot be scheduled using a regular slot, we assume all the flows in our system are being transmitted under small slots. Then, we combine the conjoint slots for the flows that cannot be transmitted under a small slot. We called this method transmission matching (TM).

Fig. 3 is an example of TM where the packet in f_2 is smaller than that in f_1 and can be transmitted under a small slot. Initially, we allocate resources under a regular slot; consequently, f_2 would miss its deadline, as shown by the red labels. To improve the schedulability, we assign slots under our small slot. As shown in Fig. 3(c), f_1 needs two continuous small slots for each timed transmission; in addition, the transmission speed of f_2 is increased: (f_2 can be scheduled and reach its destination at the third regular slot).

Lemma 1. *When we ignore the lower bound of the packet size, the system that satisfies our heterogeneous frame (all small slots are the same as l, and $\beta = \frac{l}{L}$) can achieve the same performance regardless of the β_i value. That is, $\beta^w = \beta_1\beta_2\ldots\beta_w,\ w \to \infty$, where $1 > \beta_1 > \beta_2 > \ldots\beta_w > 0$.*

Proof. When we ignore the lower bound of the packet size, the slot lengths can be arbitrarily decreased based on the packet sizes. As the number of types of slots n increases, a regular slot is divided into an increasing number of small small slots. Then, the issue becomes a math problem of comparing the limits of β^w and $\beta_2\ldots\beta_w$ when $w \to \infty$. Because $1 > \beta > 0$ and $1 > \beta_1 > \beta_2 > \ldots\beta_w > 0$, we can obtain the limits as follows:

$$\lim_{w\to\infty}(\beta^w) = 0 = \lim_{w\to\infty}(\beta_1\beta_2\ldots\beta_w). \quad (1)$$

Hence, when we ignore the lower bound of the packet size, the system using our heterogeneous slots can obtain the same performance regardless of the value of β (β_i). That is, $\beta^w = \beta_1\beta_2\ldots\beta_w$, $w \to \infty$, where $1 > \beta_1 > \beta_2 > \ldots\beta_w > 0$. □

However, in real-world applications, we cannot ignore the packet size boundary (5 to 320 Bytes). In many flows, the packet sizes are considerably smaller than 320 bytes (for example, a packet of temperature information usually requires only (5+1) bytes). We assume that the length of the smallest packet in the system is 10 bytes, then we can obtain $\frac{10}{320} = 0.5^5$; hence, β is $\exp(0.5, 5)$, $w = 5$. In this case, the small slot is $l = L \times \beta_5 = 10 \times 0.03125 = 0.3125$ ms. Considering the practicality of our system and the accuracy of time synchronization, we cannot infinitely divide a regular slot (the accuracy of time synchronization in industrial wireless system is μs class, and in this case, the packet is larger than the time-synchronization accuracy [38]). When we assume the lower bound of a small slot is $l^{low} = \beta^w * L$, the system has the same performance regardless of whether the small slot is set to $\beta_1, \beta_2 \dots \beta_w$ or $\beta^w = \frac{l^{low}}{L}$. Hence, we can extend Lemma 1 as follows.

Hypothesis 1. Considering the practicality of our system and the accuracy of time synchronization, the system has the same performance regardless of whether we set the small slot to $\beta_1, \beta_2 \dots \beta_w$ or $\beta^w = \frac{l^{low}}{L}$. We can set the small slot to l^{low} because the system satisfies our heterogeneous-slot model regardless of the value of β (or β_i).

5. Split scheduling algorithm

In this section, we propose a split scheduling algorithm (SSA) for the heterogeneous-slot IWSN. The key idea of SSA is that small slots for the flows are reallocated when the system cannot be scheduled. In this work, SSA considers only the situation in which there are two types of slots; however, it is easy to extend the types of slots by Lemma 1. The precise description of SSA in one frame is shown in Algorithm 1.

Algorithm 1 Split Scheduling Algorithm.

```
Require: F;
Ensure: the schedulability of the system;
1: allocate resources for each flow f_i by the RM scheduling algorithm
2: if the system cannot be scheduled then
3:    split regular slots into small slots and reallocate again;
4:    if the system cannot be scheduled then
5:       return failed;
6:    else
7:       return succeed;
8:    end if
9: end if
```

The split scheduling algorithm is very simple and can be described briefly as follows.

- We first allocate network resources through the RM scheme for all flows. Then, we judge whether or not the system can be scheduled.
- If the system cannot be scheduled, we split the regular slots into smaller small slots by β. Then, we reallocate the network resources using RM and reassess the schedulability; otherwise, we return a success status.

By compacting the transmission, SSA makes an unscheduled system feasible, such as the example in Fig. 3. By reducing the length of the transmission slot, the system acquires more slot resources. For a system that cannot be scheduled due to transmission conflicts, SSA can make better use of the channel resources to enhance the system's schedulability. However, SSA does not apply to all transmission conflict situations.

Proposition 1. *Because not all packets can be split, we cannot guarantee schedulability by SSA even if we infinitely split the slots. Furthermore, SSA cannot improve the performance for all systems even if we could infinitely split the slots and there were a sufficient number of available channels.*

Proof. We prove this by contradiction. As Fig. 4 shows, f_1 generates small packets that can be infinitely sliced. The path of f_1 is $node1 \to node2 \to node3 \to node0$. Moreover, f_2 and f_3 generate regular packets that cannot be split, and their paths are $node4 \to node2 \to node3 \to node0$ and $node5 \to node3 \to node0$, respectively. The priority of these flows is assigned by RM; therefore, f_3 has the highest priority, and f_1 has the lowest priority. Initially, neither f_1 nor f_2 can be scheduled as shown by the red labels. Therefore, SSA slices the regular slots into small slots. However, f_1 and f_2 still cannot be scheduled (actually, f_1 could be scheduled if we were to reverse the priority of f_1 and f_2) but that contradicts the given condition.

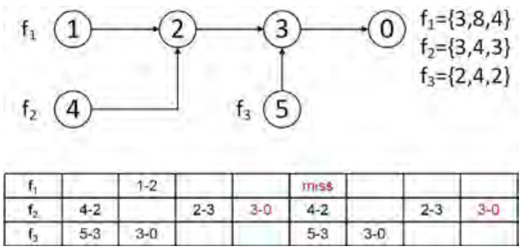


Fig. 4. An example of the disabled situation of SSA.

Hence, SSA cannot improve the performance for all systems even if we could infinitely split the slots and there were a sufficient number of available channels. □

6. Improving transmission efficiency

In Section 5, we proposed SSA, which provides a possible approach for improving system schedulability. However, the packet transmission under a small slot may be delayed by having to wait for a regular slot (because the node is occupied by a high-priority regular packet) causing it to miss deadline, as shown in Fig. 4. Hence, we study (1) the point at which the performance must be improved and (2) how to improve the transmission efficiency by avoiding unnecessary waits. The analysis begins with $\beta = 0.5$, $w = 1$.

The root cause of waiting is transmission conflicts of the regular flows in F^L. That is, as Fig. 4 shows, f_1 and f_2 transmit in our heterogeneous-slot system, but f_1 is delayed at node2 because node2 is already occupied by f_2 at the first slot. To address this issue, we introduce utilization and propose a double plug-in algorithm (DPA) to further improve system schedulability.

The utilization bound is a well-known concept first introduced by Liu and Layland in [39] and is a good metric for evaluating the "quality" of a scheduling algorithm. The utilization of node u_i is defined as follows.

Definition 3. When j flows are transmitted through node i, $j \neq 0$, the utilization of node i in a frame u_i can be expressed as $u_i = \frac{\tau_i}{T}$, where τ_i is the total number of transmission slots in node i during T (the transmission contains sending and receiving). Then, the system utilization is $U = \sum_{i=1}^{e} u_i$, where e is the number of nodes in the system.

When there are j flows transmitting through node i, its utilization can be expressed as follows:

$$u_i = \frac{\tau_i}{T} = \frac{\sum_{k=0}^{j} \frac{2T}{t_k}}{T} = \sum_{k=0}^{j} \frac{2}{t_k}. \tag{2}$$

Considering the utilization characteristics (in single-processor real-time system scheduling, utilization satisfies $u_j^* = \frac{c_j}{t_j}$, $U^* = \sum u_j^* \leq 1$, where u_j^* is the utilization for flow j and U^* is the sum utilization of all the flows in the system), we can derive the following result:

Theorem 1. *The system cannot be scheduled when any of the following conditions are satisfied: (1)* $\sum_{k=0}^{j} \frac{2}{t_k} > 1$*, where j is the flow transmission through node i or (2)* $\sum_{i=1}^{e} u_i > m$*.*

Proof. According to Eq. (2) we can obtain the utilization of node i as $u_i = \frac{\tau_i}{T}$. We also know that each slot can be transmitted through most one node a time. Hence, no slot can be used when the utilization satisfies $\sum_{k=0}^{j} \frac{2}{t_k} > 1$. Thus, the system cannot be scheduled when a node satisfies $\sum_{k=0}^{j} \frac{2}{t_k} > 1$, where j is the number of flows transmitted through node i.

Due to the phenomenon of channel contention in the TDMA network, each channel is assigned to at most one transmission for one slot. When the number of concurrent transmissions is larger than the number of channels, the low-priority flows will miss their deadlines. In other words, the system cannot be scheduled when $\sum_{i=1}^{e} u_i > m$.

Hence, the system cannot be scheduled when it satisfies any of the following conditions: (1) $\sum_{k=0}^{j} \frac{2}{t_k} > 1$, where j is the number of flows transmitting through node i and (2) $\sum_{i=1}^{e} u_i > m$. □

In addition, the relationship between u_i and u_h^* is as follows:

$$u_i = \sum_{k=0}^{j} \frac{2}{t_k} = \sum_{k=0}^{j} \frac{2}{c_k} \frac{c_k}{t_k} = \sum_{k=0}^{j} \frac{2u_k^*}{c_k}. \tag{3}$$

From Eq. (2), we can see that the denominator is the same for all u_i. Then, we can obtain another result as described below.

Lemma 2. *For flow f, the node with the smallest utilization on its path is k (where k is not the source or destination node). Then, node i is an intersection node when* $u_i > u_k$ *for any node i in the path of flow f.*

Proof. Because the denominator of each node utilization is T and τ_i is the total number of transmission slots on node i, we can use u_i to express the transmission times. For any flow f, each node other than the source and destination nodes has the same utilization when we ignore the other flows. Therefore, any node with a larger utilization must be an intersection node. □

Because the transmissions are not independent on each node, we define f_j's worst-case transmission time on node i as d_j^i (f_j cannot be scheduled when it is sent after d_j^i). Then, the second challenge in Section 3.2 involves how to satisfy each d_j^i without affecting the other flows.

We explain this with an example. In Fig. 5, when there are three flows transmitting through node i, the deadlines and periods of these flows are {2, 2}, {2, 4} and {3, 4}. We can enhance the transmission speed of f_2 and f_3 by SSA. Without changing the system transmission scheme, we can guarantee the schedulability by using small slots. However, SSA cannot guarantee the schedulability when reducing d_2^i and d_3^i to 1 (actually, the system can be scheduled by plugging f_2 and f_3 into the first slot).

We denote $t^{current}$ as the current time, $0 \le t^{current} \le T$. Then, we can obtain the transmission time of the current packet as

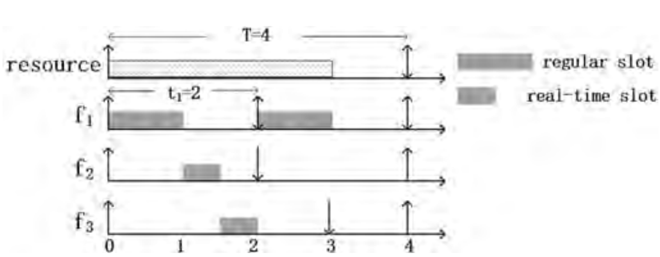


Fig. 5. Relationship between node utilization and transmission.

$d_j^{current} = d_j - t^{current} \bmod t_j$. When i is the kth node on the path of F_j, we obtain d_j^i as follows:

$$d_j^i = \begin{cases} d_j^{current} - (c_j - k) & d_j^{current} \ge c_j - k \\ 0 & d_j^{current} \le c_j - k \end{cases}, \tag{4}$$

Similarly, as d_j^i, we denote the earliest arrival time at which the current packet of f_j can reach node i as a_j^i, $a_j^i = k, i = p_j^k$. Without transmission delay, the packet can reach its destination when it can be transmitted between a_j^i and d_j^i, $i \in p$. Obviously, when $d_j^{current} = d_j$, the length of each transmission window for the nodes on f_j is the same and is equal to $\Delta_j = d_j^i - a_j^i = d_j - (c_j - k) - k = d_j - c_j$. When the allocated slot of node i on the path of the unscheduled flow f_j is denoted as s_j^i, assuming the number of flows is no larger than the number of channels, we can obtain the following results:

Theorem 2. *In one period of any flow j that cannot be scheduled, if we can find a previous slot* $a_j^i \le S_j^i < s_j^i, i \in p_j$ *that has not been allocated for all nodes on the path of* f_j *and the previous slots always satisfy* $S_j^{p_j^i} < S_j^{p_j^k}, i < k$*, we can improve the performance of* f_j*. We call these previous slots idle slots.*

Proof. Transmission conflicts are eliminated when there idle slots exist for each node on the path of f_j. Because each slot can transmit in one hop, the packet can be transmitted from the source to its destination when the idle slots satisfy $a_j^{p_j^k} > S_j^{p_j^i}, i < k$. That is, f_j is schedulable when we can find a previous slot $a_j^i \le S_j^i < s_j^i, i \in p_j$ that has not been allocated for all nodes on the path of f_j and where the previous slots always satisfy $S_j^{p_j^i} < S_j^{p_j^k}, i < k$. □

Based on Theorem 2, we propose DPA to allocate slots to improve the schedulability of the system. The pseudocode of DPA is shown in Algorithm 2.

DPA is a method that improves system schedulability by accelerating the flow transmission speed with two-stage plug-in idle slots. DPA involves three parts. In the first part (lines 1 to 3), DPA splits the slots as in SSA and sorts the nodes on each flow path that cannot be scheduled under SSA in descending order. In the second part (lines 4 to 24), DPA begins to accelerate the flow transmission speed for $F^{l'}$ and $F^{L'}$, respectively. For $f_{l'}$, because the available time window for node i on the path of $f_{l'}$ is $[a_{l'}^i, a_{l'}^i + \Delta_{l'})$, DPA begins to search for idle small slots in $[a_{l'}^i, t_{l'}^i)$. When an idle but usable small slot exists that $f_{l'}$ can use, DPA will adjust the allocation of $f_{l'}$. The speed-up method for $f^{L'}$ is similar to that for $f^{l'}$. Finally, DPA judges the system's schedulability under DPA (lines 25 to 29). The time complexity of DPA is $O(ze)$, where z is the number of flows, and e is the number of nodes.

Theorem 3. *DPA has better performance than does SSA; that is, the system cannot be scheduled under SSA when it is unschedulable under DPA.*

Proof. Because DPA can speed up transmissions detecting and using idle slots, the unscheduled flows under SSA have the opportunity to meet their deadline through DPA. Hence, DPA achieves better performance than SSA; that is, the system cannot be scheduled under SSA when it is unschedulable under DPA. □

7. Experimental results

In this section, we describe the simulations and experiments conducted to evaluate the performance of our proposed methods. To illustrate the variation tendency of F^l, we define the proportion

Algorithm 2 Double Plug-in Algorithm.

Require: F;
Ensure: the schedulability of the system;
SSA;
find the flows that cannot be scheduled, and then join these flows in flow sets $F^{l'}$ and $F^{L'}$;
calculate the node utilizations on the paths of these flows, and then join the nodes in the corresponding node sets $N^{l'}$ and $N^{L'}$ in descending order;
sort $F^{l'}$ and $F^{L'}$ by their priorities in descending order, respectively;
for l=0 to the length of $F^{l'}$ **do**
  for each node in $N^{l'}$ **do**
    search the idle small slot for $f_{l'}$ from $[a^i_{l'}]$ to the current allocation slot;
    if there is an idle small slot that satisfies Theorem 2 **then**
      plug-in the idle small slot and release the original slot;
    else
      break;
    end if
  end for
end for
for L=0 to the length of $F^{L'}$ **do**
  for each node in $N^{L'}$ **do**
    search the idle slot for $f_{L'}$ from $[a^i_{L'}]$ to the current allocation slot;
    if there is an idle slot that satisfies Theorem 2 **then**
      plug-in the idle slot and release the original slot;
    else
      break;
    end if
  end for
end for
if all the flows can reach their destinations before their deadlines **then**
  return Schedulable;
else
  return Unschedulable;
end if

Table 1
Simulation parameters.

Parameter	Description
e	The number of nodes
z	The number of flows
U	System utilization
u_i	Node i's utilization
β	The proportion of real-time and traditional slots
N	Node set
F	Flow set
L	The length of regular slot
l	The length of small slot
B	The proportion of real-time flows in F

of real-time flows in F as $B = \frac{F^l}{F}$. The constants used in this section are $L = 10$ ms, $l = 5$ ms and $\beta = 0.5$. All algorithms are implemented in C language. These programs run on a Windows machine with 3.4 GHz CPU and 8GB memory. Some simulation parameters are summarized in Table 1.

7.1. Simulation

We evaluate our proposed heterogeneous-slot algorithms SSA and DPA using the RM scheduling policy under a traditional slot model (RM is widely used in IWSN; other policies such as Earliest Deadline First are usually used in CPU scheduling). To illustrate the applicability of our approaches, we randomly generate several test cases for each parameter configuration. For each test case, the network gateway is placed at the center of the deployment area A, and the other nodes are randomly deployed around the gateway. When setting the transmission range as 40 m, the number of nodes e and the playground area A should satisfy $\frac{e}{A} = \frac{2\pi}{40^2 \times \sqrt{27}}$. Two nodes can communicate with each other when the distance between each other is less than 40 m; they are adjacent nodes. We can determine the transmission path by randomly connecting the nodes from each source node to the destination. If some source nodes cannot be connected to the destination, their locations are randomly generated again. In each configuration, system performance is evaluated by the acceptance ratio, which is the proportion of schedulable flows in F. The relationships among the number of nodes, the transmission range, and the deployment area is set according to the suggestion in [40].

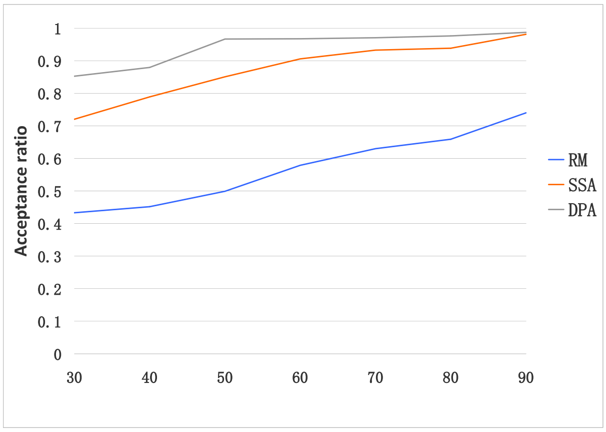
Fig. 6(a) shows the relationship between the acceptance ratio and the number of nodes when the number of channels is $m = 16$, the number of flows is $z = 15$, proportion of real-time flows in F is $B = 0.25$, and system utilization is $U^* = 0.2$. The acceptance ratios under different scheduling policies satisfy $DPA > SSA > RM$, and the results all increase as e increases, which is the same as accepted. Furthermore, the acceptance ratios of SSA and DPA are almost equal when $e = 90$. That is, the number of intersection nodes decreases as e increases; consequently, the probability of plug-ins decreases. In addition, reducing the number of intersection nodes reduces transmission conflicts, which increases the SSA acceptance ratio. Hence, the acceptance ratios of SSA and DPA are basically identical.

Fig. 6(b) shows the relationship between the acceptance ratio and number of flows when the number of channels is $m = 16$, the number of nodes is $e = 50$, the proportion of real-time flows in F is $B = 0.25$, and system utilization is $U^* = 0.2$. In contrast to Fig. 6(a), the acceptance ratios decrease as the number of flows increase because the increasing workload causes more conflicts. However, the reduction under SSA and DPA is much slower than under RM because the small slots release more unoccupied resources, thus improving resource utilization. Because DPA achieves maximal resource utilization, it performs better than the other two algorithms.

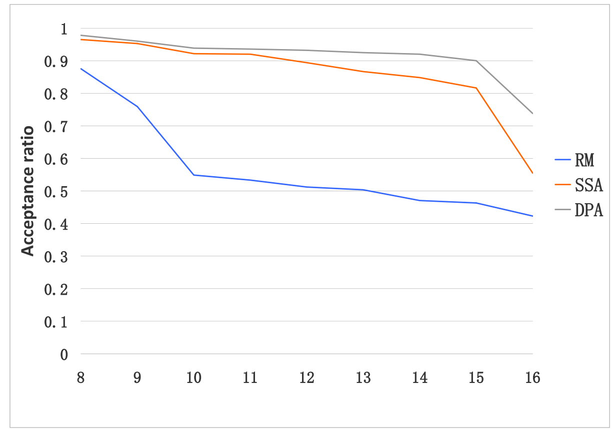
Fig. 6(c) shows the relationship between the acceptance ratio and number of channels when the number of nodes is $e = 50$, the number of flows is $z = 15$, the proportion of real-time flows in F is $B = 0.25$, and system utilization is $U^* = 0.2$. The acceptance ratios are proportional to m. Furthermore, the performance of DPA improves the fastest, especially in the 10 to 12 range, because the conflicts in RM and SSA restrict them from achieving better performances, whereas DPA's use of addition channels improves the system performance for idle slots. Even when the number of channels exceeds 12, the acceptance ratio of DPA still increases slowly and maintains a high acceptance ratio (above 90%).

Fig. 6(d) shows the relationship between the acceptance ratio and the proportion of real-time flows in F when the number of channels is $m = 16$, the number of nodes is $e = 50$, the number of flows is $z = 15$, and system utilization is $U^* = 0.2$. Because RM does not consider small slots, only SSA and DPA are assessed in this condition. Fig. 6(d) illustrates that as the proportion of real-time flows increases in F, the acceptance ratios of SSA and DPA gradually converge because DPA gradually degenerates into SSA when a significant number of flows can be transmitted under small slots.

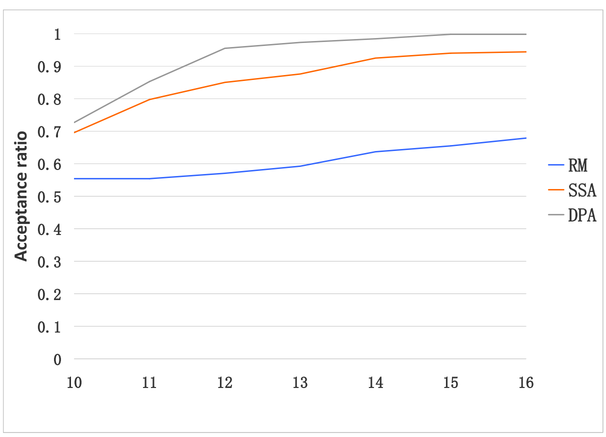
Fig. 6(e) shows the relationship between the acceptance ratio and system utilization U^* when the number of channels is $m = 16$, the number of nodes is $|N| = 50$, the number of flows is $z = 15$, and the proportion of real-time flows in F is $B = 0.25$. When



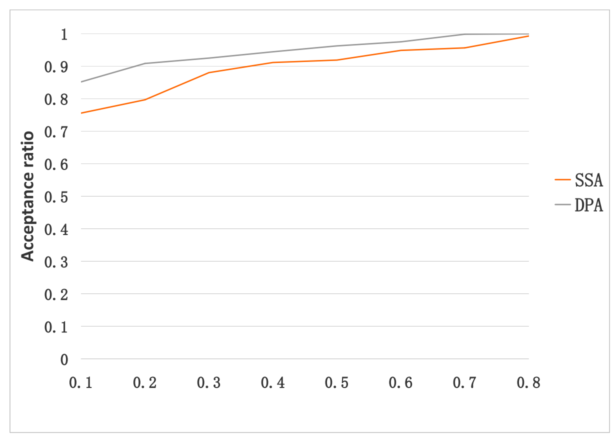
(a) Changing the number of nodes.



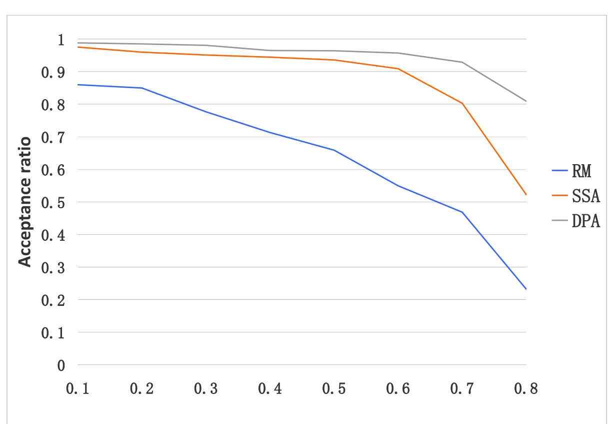
(b) Changing the number of flows.



(c) Changing the number of channels.

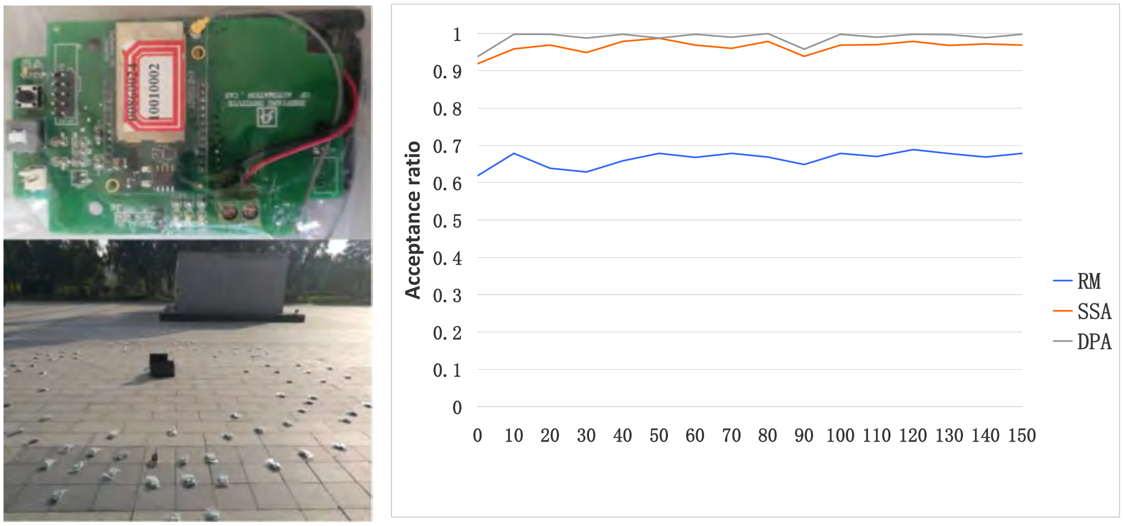


(d) Changing B.



(e) Changing system utilization U^*.

Fig. 6. The relationship between the acceptance ratio and system parameters.



(a) Our real testbed, (b) System performance,

Fig. 7. Our real testbed.

system utilization changes from 0.1 to 0.8, the performances of all three policies decrease because the probabilities of transmission conflicts and channel contention increase with increasing utilization. Initially (0.1 to 0.6), DPA achieves stable performance because it can use the idle slots; however, when the utilization exceeds 0.6, the number of idle slots decreases rapidly and the performance of DPA declines.

7.2. Experiment

We deployed 80 nodes in Hun River park, as shown in Fig. 7(a). Our network nodes are equipped with MSP430 and CC2420 chips. For this deployment, we compared the acceptance ratios under RM, SSA and DPA, where the parameters are $m = 16$, $z = 15$, $B = 0.25$, and $U^* = 0.2$. We first generate the schedules under different policies, then we download the schedules to each node, respectively. The length of slot can be adjust by changing the sampling frequency of the clock cycle. Because 81 nodes are considered, the program download takes a considerable amount of time. The network gateway is placed at the center, and the other nodes are divided into 8 clusters deployed around the gateway. Each cluster includes 1 cluster head and 9 child nodes, and the number of hops for each flow is between 3 to 5. The acceptance ratio is the proportion of schedulable flows in F, as shown in Fig. 7(b), where the x-axis represents our test time (150 min) and each point in Fig. 7(b) is the average acceptance ratio over 10 min under each scheduling policy (the number of packets received under each policy divided by the total number of packets sent during the 10 min period). The average acceptance ratios of RM, SSA and DPA are 66.4%, 96.4% and 98.8%, respectively. Fig. 7(b) illustrates that the network's performance can be substantially improved by splitting regular slots into small slots. Moreover, the system acceptance ratio can be further enhanced under DPA using idle slots.

8. Conclusion

The existing TDMA protocol assigns slots with a fixed length, which wastes network resources even when the system is unschedulable. This paper proposes a heterogeneous framework to improve network resource utilization by splitting the regular slots into small slots. We first study the transmission-matching problem and prove that our heterogeneous-slot framework can be used when flows with arbitrary packet sizes exist. Then, we study the resource allocation problem and propose two scheduling algorithms, SSA and DPA. We perform extensive simulations and realistic testbed experiments. The results confirm the efficiency of our methods. Furthermore, the experiments show that DPA can improve the acceptance ratio by approximately 48.8% compared to the existing RM. In future work, we will further analyze the TDMA header and extend our research to multilayered systems. Furthermore, by analyzing the characteristics of different scheduling methods, we will study how to improve DPA, and propose a serious variable slot scheduling policies based on DPA, such as DPA-DM, DPA-LLF. We also plan to study the lifetime of heterogeneous-slot IWSNs.

Acknowledgments

This work was partially supported by the National Science and Technology Major Project (2017ZX02101007-004), the National Natural Science Foundation of Liaoning province (20180520029 and 20180540114), the National Natural Science Foundation of China (61803368), the independent subject of State Key Laboratory of Robotics, and the Youth Innovation Promotion Association CAS.

Conflicts of interest

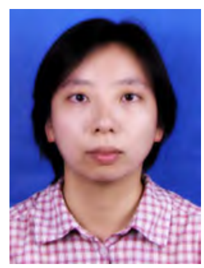
None.

References

[1] A. Fisher, C.A. Jacobson, E.A. Lee, R.M. Murray, A. Sangiovanni-Vincentelli, E. Scholte, Industrial Cyber-Physical Systems iCyPhy, Springer International Publishing, 2014.
[2] J. Heo, J. Hong, Y. Cho, EARQ: energy aware routing for real-time and reliable communication in wireless industrial sensor networks, IEEE Trans. Ind. Inf. 5 (1) (2009) 3–11, doi:10.1109/TII.2008.2011052.
[3] H. Ramamurthy, B.S. Prabhu, R. Gadh, A.M. Madni, Wireless industrial monitoring and control using a smart sensor platform, IEEE Sens. J. 7 (5) (2007) 611–618, doi:10.1109/JSEN.2007.894135.
[4] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, W. Pratt, WirelessHART: applying wireless technology in real-time industrial process control, in: 2008 IEEE Real-Time and Embedded Technology and Applications Symposium, 2008, pp. 377–386, doi:10.1109/RTAS.2008.15.
[5] S. Petersen, S. Carlsen, WirelessHART versus ISA100.11a: the format war hits the factory floor, IEEE Ind. Electron. Mag. 5 (4) (2011) 23–34.
[6] M. Zheng, W. Liang, H. Yu, Y. Xiao, Performance analysis of the industrial wireless networks standard: wia-pa, Mob. Netw. Appl. 22 (1) (2017) 139–150, doi:10.1007/s11036-015-0647-7.

[7] J. Yao, X. Xu, X. Liu, MixCPS: mixed time/event-triggered architecture of cyber physical systems, Proc. IEEE 104 (5) (2016) 923–937.
[8] F. Nawaz, V. Jeoti, Performance assessment of WirelessHART technology for its implementation in dense reader environment, Computing 98 (3) (2016) 257–277.
[9] C. Xia, J. Xi, Z. Peng, Resource analysis for wireless industrial networks, in: International Conference on Mobile Ad-Hoc and Sensor Networks, 2016, pp. 424–428.
[10] S. Wei, M. Gidlund, F. Dobslaw, SAS-TDMA: a source aware scheduling algorithm for real-time communication in industrial wireless sensor networks, Wirel. Netw. 19 (6) (2013) 1155–1170.
[11] S. Wei, T. Zhang, F. Barac, M. Gidlund, Prioritymac: a priority-enhanced mac protocol for critical traffic in industrial wireless sensor and actuator networks, IEEE Trans. Ind. Inf. 10 (1) (2013) 824–835.
[12] A. Willig, Recent and emerging topics in wireless industrial communications: a selection, IEEE Trans. Ind. Inf. 4 (2) (2008) 102–124, doi:10.1109/TII.2008.923194.
[13] T.M. Chiwewe, C.F. Mbuya, G.P. Hancke, Using cognitive radio for interference-resistant industrial wireless sensor networks: an overview, IEEE Trans. Ind. Inf. 11 (6) (2015) 1466–1481, doi:10.1109/TII.2015.2491267.
[14] J. AraÃºjo, M. Mazo, A. Anta, P. Tabuada, K.H. Johansson, System architectures, protocols and algorithms for aperiodic wireless control systems, IEEE Trans. Ind. Inf. 10 (1) (2014) 175–184, doi:10.1109/TII.2013.2262281.
[15] F. Dobslaw, T. Zhang, M. Gidlund, End-to-end reliability-aware scheduling for wireless sensor networks, IEEE Trans. Ind. Inf. 12 (2) (2016) 758–767.
[16] H. Farag, E. Sisinni, M. Gidlund, P. Osterberg, Priority-aware wireless fieldbus protocol for mixed-criticality industrial wireless sensor networks, IEEE Sens. J. PP (99) (2018). 1–1
[17] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, Y. Chen, Real-time wireless sensor-actuator networks for industrial cyber-physical systems, Proc. IEEE 104 (5) (2016) 1013–1024, doi:10.1109/JPROC.2015.2497161.
[18] B.M. Wilamowski, J.D. Irwin, Industrial communication systems, 2016.
[19] X. Li, D. Li, J. Wan, A.V. Vasilakos, C.-F. Lai, S. Wang, A review of industrial wireless networks in the context of industry 4.0, Wirel. Netw. 23 (1) (2017) 23–41, doi:10.1007/s11276-015-1133-7.
[20] S.C. Ergen, P. Varaiya, TDMA scheduling algorithms for wireless sensor networks, Wirel. Netw. 16 (4) (2010) 985–997.
[21] A. Saifullah, Y. Xu, C. Lu, Y. Chen, Real-time scheduling for WirelessHART networks, in: 2010 31st IEEE Real-Time Systems Symposium, 2010, pp. 150–159, doi:10.1109/RTSS.2010.41.
[22] A. Saifullah, Y. Xu, C. Lu, Y. Chen, End-to-end communication delay analysis in industrial wireless networks, IEEE Trans. Comput. 64 (5) (2015) 1361–1374, doi:10.1109/TC.2014.2322609.
[23] X. Jin, F. Kong, L. Kong, H. Wang, C. Xia, P. Zeng, Q. Deng, A hierarchical data transmission framework for industrial wireless sensor and actuator networks, IEEE Trans. Ind. Inf. 13 (4) (2017) 2019–2029, doi:10.1109/TII.2017.2685689.
[24] D. Yang, Y. Xu, H. Wang, T. Zheng, H. Zhang, H. Zhang, M. Gidlund, Assignment of segmented slots enabling reliable real-time transmission in industrial wireless sensor networks, IEEE Trans. Ind. Electron. 62 (6) (2015) 3966–3977, doi:10.1109/TIE.2015.2402642.
[25] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial internet of things: challenges, opportunities, and directions, IEEE Trans. Ind. Inf. PP (99) (2018). 1–1
[26] C. Xia, X. Jin, L. Kong, P. Zeng, Bounding the demand of mixed-criticality industrial wireless sensor networks, IEEE Access 5 (2017) 7505–7516, doi:10.1109/ACCESS.2017.2654483.
[27] D. Gunatilaka, M. Sha, C. Lu, Impacts of channel selection on industrial wireless sensor-actuator networks, in: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1–9, doi:10.1109/INFOCOM.2017.8057049.
[28] H. Zhang, P. Gburzynski, A variable slot length TDMA protocol for personal communication systems, Wirel. Pers. Commun. 22 (3) (2002) 409–432.
[29] S. Hussain, A.S. Zahmati, X. Fernando, LASA: low-energy adaptive slot allocation scheduling algorithm for wireless sensor networks, in: Sarnoff Symposium, 2009. SARNOFF '09. IEEE, 2009, pp. 182–187.
[30] S.B. Scherzer, P.A. Worfolk, A.D. Haken, S. Ponnuswamy, R. Vainish, Implementing all wireless network over WiFi equipment using scheduled TDMA, 2015, US Patent 9,179,495.
[31] E. Wandeler, L. Thiele, Optimal TDMA time slot and cycle length allocation for hard real-time systems, in: Design Automation, 2006. Asia and South Pacific Conference on, 2006, pp. 479–484.
[32] C. Xia, X. Jin, L. Kong, P. Zeng, Scheduling for emergency tasks in industrial wireless sensor networks, Sensors 17 (7) (2017).
[33] S. Yao, S. Fu, J. Li, M. Tang, P. Shum, D. Liu, Phase noise tolerant inter-carrier-interference cancellation for WDM superchannels with sub-Nyquist channel spacing, Opt. Express 21 (18) (2013) 21569–21578.
[34] P. Kyasanur, N.F. Vaidya, Routing and interface assignment in multi-channel multi-interface wireless networks, in: Wireless Communications & Networking Conference, 2005.
[35] International Electrotechnical Commission, IEC 62591, Ed. 2.0: Industrial Communication Networks-Wireless Communication Network and Communication Profiles-WirelessHART (Draft)[J], 2013.
[36] IEEE, Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans), IEEE Standard for Information Technology (2003).
[37] J. Xi, F. Kong, L. Kong, L. Wei, Z. Peng, Reliability and temporality optimization for multiple coexisting WirelessHART networks in industrial environments, IEEE Trans. Ind. Electron. PP (99) (2017). 1–1
[38] F. Wang, P. Zeng, H. Yu, Y. Xiao, Error compensation algorithm in wireless sensor networks synchronisation, Int. J. Sens. Netw. 10 (3) (2011) 123–131.
[39] C.L. Liu, J.W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment, J. ACM (JACM) 20 (1) (1973) 46–61.
[40] T. Camilo, J.S. Silva, A. Rodrigues, F. Boavida, Gensen: a topology generator for real wireless sensor networks deployment, in: Ifip Wg 10.2 International Conference on Software Technologies for Embedded and Ubiquitous Systems, 2007, pp. 436–445.

Changqing Xia received the Ph. D. degree from Northeastern University, China in 2015. He is currently an assistant professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include wireless sensor networks and real-time systems, especially the real-time scheduling algorithms, and smart energy systems.



Xi Jin received the Ph. D. degree from Northeastern University, China in 2013. She is currently an associate professor at Shenyang Institute of Automation, Chinese Academy of Sciences. Her research interests include wireless sensor networks and real-time systems, especially the real-time scheduling algorithms, and worst case end-to-end delay analysis.

Linghe Kong is currently an associate professor with Department of Computer Science and Engineering at Shanghai Jiao Tong University. Before that, he was a postdoctoral researcher at Columbia University, McGill University, and Singapore University of Technology and Design. He received his Ph.D. degree from Shanghai Jiao Tong University, 2012, his Master degree from TELECOM SudParis 2007, and his B. E. degree from Xidian University, 2005. His research interests include wireless communication, sensor networks, mobile computing, Internet of things, and smart energy systems.

Chi Xu received the Ph.D. degree from University of Chinese Academy of Sciences, China, in 2017. He has been with Shenyang Institute of Automation, Chinese Academy of Sciences, China, since 2013, and currently is serving as an assistant professor. He is a member of IEEE, China Computer Federation (CCF), and Chinese Association of Automation (CAA). He also serves as a 3GPP Delegate for RAN1. His research interests include cognitive radio networks, wireless sensor networks, industrial internet, and 5G.

Peng Zeng received his Ph. D. degree from Shenyang Institute of Automation, Chinese Academy of Sciences. He is currently a professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include industrial communication and wireless sensor networks.